Monte Carlo Methods Are Perfectly Situated to Enable Exascale Scientific Computing

White Paper: DOE Applied Mathematics Research for Exascale Computing

Michael Mascagni (`mascagni@fsu.edu`) and Michela Taufer (`taufer@udel.edu`)

Departments of Computer Science, Florida State University and University of Delaware

Introduction

Monte **C**arlo **m**ethods (MCMs) are based on the simulation of stochastic processes whose expected values are equal to computationally interesting quantities. MCMs offer simplicity of construction, and are often designed to mirror some process whose behavior is only understood in a statistical sense. There are a wide class of problems where MCMs are the only known computational methods of solution. For example, MCMs are the only feasible numerical approach to many-body quantum computations, a wide variety of computations in finance and risk assessment, and even in the solution of certain **p**artial **d**ifferential **e**quations (PDEs) that arise in electrostatic and magnetostatic computations. The advantage of MCMs in these problems are due to the fact that MCMs (1) can sample in high dimensions without the complexity issues that high dimensions create in deterministic methods, (2) allow for rapid evaluation of quantities of interest to low accuracy, (3) allow functionals of the solutions to field problems (think point values of the solutions to PDEs), (4) allow for the rapid, stochastic, application of linear operators, both finite and infinite dimensional, and finally (5) provide a wealth of parallelization opportunities to the underlying sampling in the method at both the fine-grain and coarse-grain level. With the advent of exascale computing, we believe that there is the need for further research into stochastic algorithms, and for applied research in the efficient implementation and creation of MCM-based mathematical software that will enable exascale computations on a wide range of significant scientific and technological problems while retaining the intrinsic parallelism, fault-tolerance, favorable computation-communication ratios, and algorithmic resilience.

Exascale Computing with MCMs

As **h**igh-**p**erformance **c**omputing (HPC) hardware is scaled up from the petascale to the exascale, many applications do not have the intrinsic structure to provide the parallelism required to productively occupy all of the computing elements that can operate simultaneously. In addition, as the number of independent computational elements increases, the resulting stability of the computing platform decreases, which introduces the need for system-level or algorithm-level resilience. MCMs handle the scalability issues along with resilience due to the nature of MCMs themselves. Moreover, MCMs are naturally parallel due to the fact that independent sampling can be distributed as one likes or needs to satisfy computational or communication requirements. The amount of communication that one does per unit of computation can be tuned in a very straight-forward fashion to optimize performance. This is due to the fact that each quantity of interest is computed with a running mean and variance, which requires only a single integer and two floating-point values of storage. This simplifies interprocessor communication, and enables a very light-weight application-level check-point of all MCMs. Besides these three stored values per quantity of interest, the only other piece of information required to restart the application is the random number generator's state. These key features of MCMs simplify interprocessor communication and enable a very light-weight application-level check-point of all MCMs. These features also allow very high level fault-tolerance to be achieved in a very generic way across MCMs. Ultimately, MCMs can tolerate otherwise catastrophic loss of data while incurring only an accuracy penalty. Thus, parts of an exascale system can fail, and if the hardware permits, the intact part of the machine can continue with the MCM and can ignore the resulting loss of data from the disabled hardware if the problem distribution is made with an eye towards partial hardware failure. The resulting computation will be slowed, but can continue and complete.

It is important to note that an essential enabling technology for the scalable execution of MCMs on modern HPC hardware is robust, reproducible, and high-quality parallel random number generation software. For distributed memory HPC hardware, the **S**calable **P**arallel **R**andom **N**umber **G**enerators ((SPRNG) library is such a library, and is the work of one of the co-authors of this white paper. However, current HPC architectures differ from the distributed-memory systems SPRNG was designed for in that they have multicore processors and perhaps multiple accelerators. Thus, a new version of a library like SPRNG that supports these architectural features is necessary.

MCM Research Requirements

There is a considerable body of applied probability and numerical stochastics that underpins MCMs for a wide variety of problems. In particular, in the West, and especially in the former Soviet Union, MCMs were developed for many of the PDEs, and their integral equation equivalents, that are commonly the core mathematical problem in the computations currently consuming the largest HPC machines. However, this work was not accompanied by stable, library-quality mathematical software. In addition, even though there is much known, the subject is not well organized, and certainly the ability for applied mathematics and computer science students to learn this material is hampered by a lack of mature, developed curriculum in this area. Thus, the first set of requirements is to organize the existing material on numerical stochastics for PDEs and integral equations from sources in the former Soviet Union, the United States and Europe. This can serve as starting point for a comprehensive curriculum to train graduate students in these methods, and to help make students who normally know only deterministic numerical methods, equally familiar with these techniques. Ultimately, this curriculum will provide a missing element in the education of students interested in uncertainty quantification, as the underlying probabilistic foundation is same.

As more computational scientists become familiar with these techniques, the need for library-quality mathematical software for solving particular problems with these MCMs is expected to grow. Thus, a major early project should be the creation of prototype mathematical libraries in a few core problem areas. The most obvious of these is in the application of MCMs to the numerical solution of elliptic PDEs. This work will require both the development of fast MCM-based solvers, and geometry packages that will support this method of solution. Since MCMs for the solution of elliptic PDEs requires the evaluation of functionals on random walks in the problem geometry, representation of complicated geometries will need to be tailored to this method of simulation. In addition, reusable libraries of probabilistic transition functions and quadrature methods should be created to enable the fast evaluation of the representational functionals.

When this is completed, the ability to solve a large class of problems will be possible with efficient and optimized MCMs. However, this can only serve as a starting point, as ways to optimize the algorithms, and to further incorporate architectural constraints into the libraries will be an ongoing effort. One should consider the development of modern numerical linear algebra libraries to appreciate the breadth of this undertaking in the MCM context.

Conclusions

The nature of MCMs makes them a robust, resilient, and naturally parallel set of algorithms for modern and future HPC platforms. They have always been highly scalable algorithms, and have provided a consistently scalable way to solve certain problems, and will continue to scale from the petascale to the exascale and beyond. The main drawbacks for the widespread use of MCMs are the development of new stochastic algorithms for a larger class of HPC-dependent problems, and starting to encapsulate these MCMs into high-quality mathematical software libraries. Addressing these drawbacks will not only impact applied mathematics and computer science development, but it will also allow the establishment of a new curriculum in stochastic numerical analysis to train numerical analysts and library developers in these qualitatively different algorithms.